

YAZILIM LABORATUVARI-1

PROJE-1

Tuba Nur AKSIN
BilgisayarMühendisliği(İÖ)
KOCAELİÜNİVERSİTESİ
Kocaeli, Türkiye
220202078@kocaeli.edu.tr

Özetçe—Bu rapor, yemek tariflerini saklayabileceği ve mevcut malzemelerle hangi yemeklerin yapılabileceğini gösteren bir masaüstü uygulaması geliştirilmesini kapsamaktadır. Uygulama, dinamik arama ve filtreleme özellikleri, veritabanı yönetimi ve kullanıcı arayüzü tasarımı ile kullanıcı dostu bir deneyim sunmaktadır. Projede, Java programlama dili kullanılacaktır. Kullanıcıların tarifleri kolayca ekleyebileceği, güncelleyebileceği ve arayabileceği bir sistem oluşturulması hedeflenmiştir.

Anahtar Kelimeler —Java, veritabanı, tarif rehberi uygulaması, normalizasyon

I. GİRİŞ,

Giriş, Günümüzde yemek tarifleri, bireylerin sağlıklı beslenme ve yemek pişirme deneyimlerini zenginleştiren önemli bir kaynaktır. Ancak, mevcut tariflerin yönetimi, kullanıcıların sahip olduğu malzemelere uygun tarifleri bulmalarını zorlaştırmaktadır. Bu bağlamda, bu proje, kullanıcıların yemek tariflerini etkili bir şekilde saklayabileceği ve sahip oldukları malzemelerle hangi yemekleri yapabileceklerini belirleyebileceği bir masaüstü uygulaması geliştirmeyi amaçlamaktadır. Uygulama, kullanıcıların tarif ekleme, güncelleme, silme, arama ve filtreleme gibi işlemleri kolayca gerçekleştirebileceği işlevlerle donatılacaktır. Ayrıca, uygulama, kullanıcı dostu bir arayüz ile desteklenerek, kullanıcıların hedeflerine minimum çabayla ulaşmalarını sağlamayı hedeflemektedir.

II. YÖNTEM

Uygulama geliştirme süreci şu aşamalardan oluşacaktır:

A. Veritabanı Tasarımı

Veritabanı, tarifler ve malzemeler için ilişkili tablolar içerecektir. Her tablonun aşağıdaki gibi tasarlanması gerekmektedir: *Tarifler Tablosu:

TarifID (Primary Key, int): Her tarif için benzersiz bir ID.

TarifAdi (varchar): Tarifin adı.

Kategori (varchar): Tarifin hangi kategoriye ait olduğunu (örneğin: AnaYemek, Tatlı).

HazırlamaSuresi (int): Tarifin hazırlanma süresi (dakika cinsinden).

Talimatlar (text): Tarifin hazırlanış adımları.

*Malzemeler Tablosu:

MalzemeID (Primary Key, int): Her malzeme için benzersiz bir ID.

MalzemeAdi (varchar): Malzemenin adı.

ToplamMiktar (varchar): Malzemenin depodaki toplam miktarı.

MalzemeBirim (varchar): Örneğin "kilo", "litre", "gram".

BirimFiyat (decimal): Malzemenin birim maliyeti.

*Tarif-Malzeme İlişki Tablosu:

TarifID (Foreign Key): İlgili tarifin ID'si.

MalzemeID (Foreign Key): İlgili malzemenin ID'si.

MalzemeMiktar (float): İlgili tarifte kullanılacak malzeme miktarı.

Normalizasyon kurallarına uygun olarak tasarlanacak ve ilişkili tablolar arasında many-to-many ilişkisi kurulacaktır.

B. Kullanıcı Arayüzü Tasarımı

Uygulamanın ana ekranında tüm tariflerin listelendiği bir alan olacaktır.

Menü seçenekleri, tarif ekleme, güncelleme ve silme işlemleri için tasarlanacaktır.

Tarif listesi, tariflerin isimleri, hazırlama süreleri ve maliyet bilgileriyle birlikte

gösterilecektir. Kullanıcı, herhangi bir tarife tıkladığında tarif detaylarını görebilecektir.

Arama ve filtreleme alanı, uygulamanın üst kısmında yer alacak ve kullanıcıların tarif arama ve filtreleme işlemlerini yapabilmesine olanak tanıyacaktır.

Sonuç ekranı, farklı kriterlere göre yapılabilecek yemekleri listeleyecektir.

C. Fonksiyonel Özellikler

*Tarif Ekleme:

Kullanıcı, tarifin adını, kategorisini, hazırlama süresini ve tarifin yapılış talimatlarını girecektir.

Her tarifte birden fazla malzeme bulunabilecektir. Kullanıcı,

malzemeleri miktarıyla birlikte ayrı ayrı girecek ve veritabanında önceden kaydedilmiş malzemeleri seçebilecektir. Yeni malzemeler için "yeni malzeme ekle" butonu ile veritabanına kaydedilebilecektir.

***Tarif Önerisi:**

Tüm tarifler içerisinde eksik malzemeli tarifler kırmızı, tüm malzemeleri yeterli olan tarifler yeşil renkte gösterilecektir. Eksik malzemeli tariflerde eksik malzemelerin toplam maliyeti hesaplanarak kullanıcıya sunulacaktır.

***Dinamik Arama:**

Arama çubuğu aracılığıyla tarif adı veya malzemeye göre arama yapılabilir. Kullanıcı, mevcut malzemeleri seçtikten sonra sistem, veritabanındaki tarifleri kontrol edecek ve eşleşme yüzdesine göre tarifleri sıralayacaktır.

***Filtreleme ve Sıralama:**

Kullanıcı, tarifleri hazırlama süresine, maliyete, malzeme sayısına, kategoriye veya maliyet aralığına göre filtreleyebilecektir.

***Tarif Güncelleme ve Silme:**

Kullanıcı, daha önce eklenen tarifleri güncelleyebilir veya silebilir. Veritabanı, otomatik olarak güncellenir.

***Duplicate Kontrolü:**

Aynı tarifin birden fazla kez kaydedilmesini önlemek için veritabanında duplicate kontrolü yapılacaktır.

D. Algoritma Geliştirme

Arama ve filtreleme algoritmaları tasarlanacak ve uygulanacaktır. Proje kapsamında, arama ve filtreleme algoritmalarının tasarımı ve uygulanması kritik bir önem taşımaktadır. Kullanıcıların, mevcut malzemelerine göre uygun tarifleri bulabilmeleri için aşağıdaki algoritmalar geliştirilecektir:

III. ALGORITMA AÇIKLAMASI

1-)Dinamik Arama Algoritması:

Amaç: Kullanıcıların tarif adı veya malzemelere göre arama yaparak uygun tarifleri hızlı bir şekilde bulmalarını sağlamak.

2-)Filtreleme Algoritması:

Amaç: Kullanıcının belirlediği kriterlere (hazırlama süresi, maliyet, kategori gibi) göre tariflerin sıralanmasını ve filtrelenmesini sağlamak.

3-)Tarif Öneri Algoritması:

Amaç: Kullanıcının mevcut malzemeleri ile eşleşen tariflerin belirlenmesi ve eksik malzemelerin maliyetinin

hesaplanması.

KAYNAKLAR

- [1] <https://medium.com/turuncu-internet-solutions/normalizasyon-nedir->
- [2] <https://www.java.com/tr/download/help/javafx.html>
- [3] <https://www.youtube.com/watch?v=8a7FTGk-vBYt=48s>

IV. KABAKOD, AKIŞ, DİYAGRAMI, UMLVE

EKRAN ÇIKTILARI

```
PROGRAM TarifUygulaması

// Veritabanı bağlantısı
CONNECT to Veritabanı

FUNCTION AddRecipe()
    INPUT recipeName, category, preparationTime, instructions
    CREATE NewRecipe in Veritabanı
    WHILE true
        INPUT ingredientName, ingredientAmount
        IF ingredientName NOT IN MalzemelerTablosu THEN
            INPUT newUnit, newUnitPrice
            CREATE NewIngredient in MalzemelerTablosu
        ENDIF
        CREATE RecipeIngredient in TarifMalzemeIliskisiTablosu
        DISPLAY "Yeni malzeme eklemek ister misiniz? (Evet/Hayır)"
        INPUT continueAdding
        IF continueAdding = "Hayır" THEN
            BREAK
        ENDIF
    ENDWHILE
    SAVE NewRecipe to Veritabanı

FUNCTION UpdateRecipe()
    INPUT recipeID
    IF recipeID IN TariflerTablosu THEN
        DISPLAY "Güncellenecek tarif bilgilerini girin."
        INPUT updatedDetails
        UPDATE Recipe in Veritabanı
    ELSE
        DISPLAY "Tarif bulunamadı."
    ENDIF

FUNCTION DeleteRecipe()
    INPUT recipeID
    IF recipeID IN TariflerTablosu THEN
        DELETE Recipe from Veritabanı
        DISPLAY "Tarif silindi."
    ELSE
        DISPLAY "Tarif bulunamadı."
    ENDIF

FUNCTION SearchRecipes()
    INPUT searchType, searchTerm
    IF searchType = "Ad" THEN
        RESULTS = FIND Recipes WHERE name LIKE searchTerm
    ELSE IF searchType = "Malzeme" THEN
        RESULTS = FIND Recipes WHERE ingredients MATCH searchTerm
    ENDIF
    DISPLAY RESULTS

FUNCTION FilterAndSortRecipes()
    INPUT criteria, order
    RESULTS = FIND Recipes ORDER BY criteria IN order
    DISPLAY RESULTS

END PROGRAM
```

```

FUNCTION filterRecipes(selectedFilter, minCost, maxCost, ingredientCount, searchTerm):
    DECLARE filteredRecipes AS LIST OF Recipe
    DECLARE sql AS STRING BUILDER WITH INITIAL VALUE "SELECT * FROM Tarifler WHERE 1=1"

    // Arama terimi filtresi
    IF searchTerm IS NOT EMPTY:
        APPEND " AND TarifAdi ILIKE ?" TO sql

    // Malzeme sayısı filtresi
    IF ingredientCount IS NOT EMPTY:
        APPEND " AND (SELECT COUNT(*) FROM TarifMalzeme WHERE TarifID = Tarifler.TarifID) = ?"

    // Maliyet filtresi
    IF minCost IS NOT EMPTY:
        APPEND " AND Maliyet ≥ ?" TO sql

    IF maxCost IS NOT EMPTY:
        APPEND " AND Maliyet ≤ ?" TO sql

    // Sıralama
    CALL appendSorting(sql, selectedFilter)

    TRY:
        // Veritabanı bağlantısını aç
        DECLARE conn AS DATABASE CONNECTION USING db.connect()
        DECLARE stmt AS PREPARED STATEMENT USING conn WITH sql

        DECLARE index AS INTEGER INITIAL VALUE 1

        // Parametreleri ayarla
        index = CALL setParameters(stmt, index, searchTerm, ingredientCount, minCost, maxCost)

        // Sorguyu çalıştır
        DECLARE rs AS RESULT SET USING stmt.executeQuery()

        WHILE rs HAS NEXT:
            ADD createRecipeFromResultSet(rs) TO filteredRecipes
    EXCEPTION SQLException:
        PRINT "Filtreleme hatası..."
        PRINT ERROR DETAILS

    // Filtrelenen tarifleri ana ekranda güncelle
    ...

```

```

FUNCTION isEmpty(value):
    RETURN value IS NOT NULL AND value IS NOT EMPTY

FUNCTION appendSorting(sql, selectedFilter):
    IF selectedFilter IS NOT NULL:
        SWITCH selectedFilter:
            CASE "Maliyet (Artan)":
                APPEND " ORDER BY Maliyet ASC" TO sql
            CASE "Maliyet (Azalan)":
                APPEND " ORDER BY Maliyet DESC" TO sql
            CASE "Hazırlama Süresi (Artan)":
                APPEND " ORDER BY HazırlamaSuresi ASC" TO sql
            CASE "Hazırlama Süresi (Azalan)":
                APPEND " ORDER BY HazırlamaSuresi DESC" TO sql

FUNCTION setParameters(stmt, index, searchTerm, ingredientCount, minCost, maxCost):
    IF searchTerm IS NOT EMPTY:
        stmt.setString(index, "%" + searchTerm + "%")
        INCREMENT index

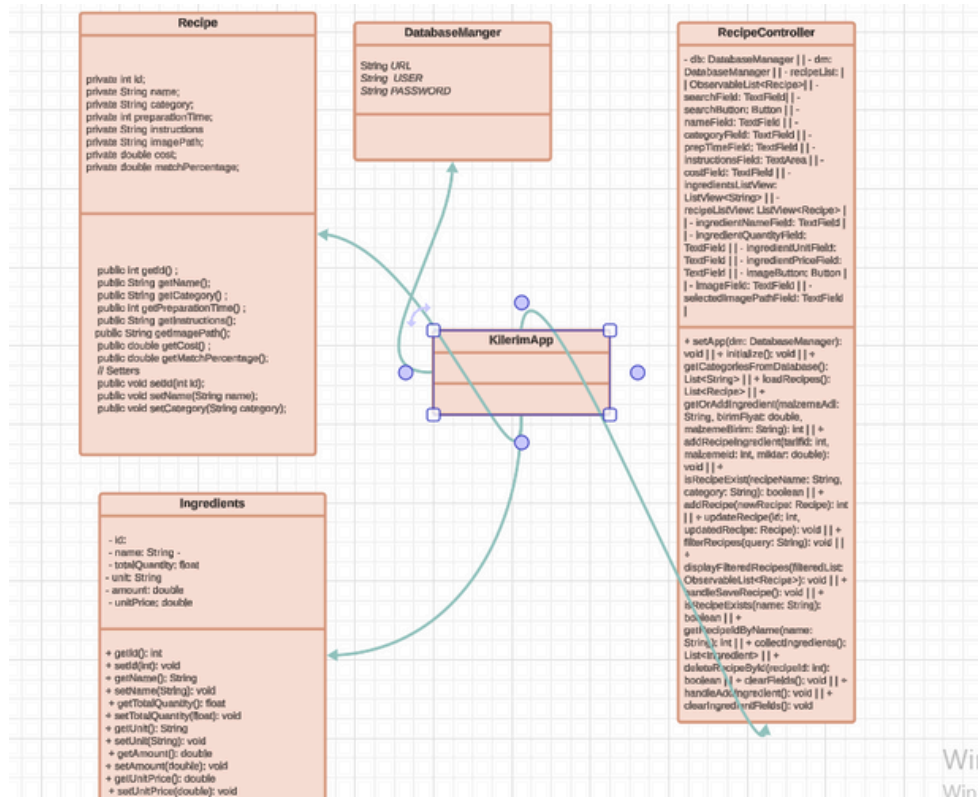
    IF ingredientCount IS NOT EMPTY:
        stmt.setInt(index, INTEGER OF ingredientCount)
        INCREMENT index

    IF minCost IS NOT EMPTY:
        stmt.setDouble(index, DOUBLE OF minCost)
        INCREMENT index


    IF maxCost IS NOT EMPTY:
        stmt.setDouble(index, DOUBLE OF maxCost)
        INCREMENT index


    RETURN index


```





AraMalzeme AraMalzemeye Göre AraFiltreleYeni Tarif Ekle



Çikolatalı Kek (0.0%)



Muzlu Puding (0.0%)



Dondurmali Brownie (0.0%)


Elmalı Turta (0.0%)


Çilekli Cheesecake (0.0%)


Muzlu Puding (0.0%)


Dondurmali Brownie (0.0%)


Elmalı Turta (0.0%)

AraMalzeme AraMalzemeye Göre AraFiltreleYeni Tarif Ekle


Maliyet (Azalan)


Maliyet Aralığı:


Malzeme Sayısı:


Uygula


Temizle



Çikolatalı Kek (0.0%)



Muzlu Puding (0.0%)



Dondurmali Brownie (0.0%)


Çilekli Cheesecake (0.0%)


Muzlu Puding (0.0%)


Dondurmali Brownie (0.0%)


Elmalı Turta (0.0%)


Çilekli Cheesecake (0.0%)

Tarif: Elmalı Turta

Kategori: Tatlı

Hazırlama Süresi: 80 dakika

Malzemeler:

Elma - 3.0 (10.0 TL)

Şeker - 400.0 (10.0 TL)


Yumurta - 3.0 (30.0 TL)

Toplam Maliyet: 4120,00 TL

Talimatlar:

elmaları soy hamurda pişir

Tarif Sil


Elmalı Turta (50.0%)